Megan Cusey

Final Project – Week 8

Summer 2019

After reviewing the K Nearest Neighbors model built back in Week 5, I’m a bit hesitant to suggest using that model in order to predict a business outcome such as meeting a 75% threshold that a customer will rate a vehicle as acceptable. On the confusion matrix, I see that k-NN was good at predicting an unacceptable rating. It had a recall of 100% for classifying true unacceptable ratings. However, I wonder how well it would do if we mapped the acceptable, good, and very good ratings as we did in this exercise. Would it help or hurt the classification?

In order to start evaluating the models generated from the auto model, I first calculated the base rate. If the model predicted unacceptable for every rating, the model would have a 70% accuracy rate. Each model should be able to attain at least a 70% accuracy rate.

The auto-model generated a series of models that optimizes parameters and model accuracy. As a result, the top three model accuracy was the Gradient Boosted Trees (99.4%), Support Vector Machine (97.8%), and Deep Learning (96.6%). The quickest performing models where the Logistic Regression and Decision Tree Models. There were no models that predicted less than the base rate, however, the accuracy percentages are eerily high. Are these results too good to be true?

Looking at ROC Comparison graph, I expected to be a little different. In the book it had specific points determined by a specific confusion matrix. However, I know that the better models have a higher true positive rate (y axis) and a lower false positive rate (x axis). Therefore, I’m looking for the model whose line is closer to the top left of the chart in this case, the Gradient Boosted Trees appeared to have the best performance.

Next to evaluate is the precision and recall rate. Similar to the above statistics, the same models appeared to have the best precision rating: Gradient Boosted Trees, Support Vector Machine, and Deep Learning. This tells me that the models have a higher rate of predicting true positive (unacceptable ratings) without making many type 1 errors (false positives). In contrast, other models had a higher recall rate indicating the ability to detect true positives in general. These models include Support Vector Machine, Deep Learning, and Logistic Regression. It’s interesting to be that Support Vector Machine and Deep Learning both have good precision and recall rates. In the given scenario, I believe we would be more concerned with precision since we want to be confident in making an offer to a customer if they are going to rate the vehicle as acceptable. We would not want a false positive and offer the vehicle to a customer and receive a sub-par rating.

Since model accuracy only considers the number of correctly labeled target values. It doesn’t consider the true positive rate (the classifier we’re most interested in) VS a true negative rate. Since the ROC Comparison graph charts the model’s accuracy as function of the true positive rate and false positive rate, the area under this curve can be used for an interesting statistic for accuracy.

Interestingly enough, the Gradient Boosted Trees model AUC is 1 followed by Support Vector Machine (.996), Deep Learning (.993), and Random Forest (.992). These are the same models that I have seen consistently throughout this analysis. Is it reasonable to believe that models can have a high overall accuracy in general and have a high true prediction accuracy? Could this be the result of overfitting the model?

To learn more about my data, it’s important to identify which attributes are most likely to signal an unacceptable rating. Most of the models agree that the most important attributes are safety and persons. Other key attributes are Luggage Boot, Purchase Price, Maintenance Costs, and Doors.

Considering all the above items, I’m still a bit skeptical of the high statistics given for models such as Gradient Boosted Trees, Support Vector Machine, and Deep Learning. These aren’t models we’ve learned about yet so are they just THAT good? I researched the auto model functionality for RapidMiner a bit and briefly looked into the processes it built. It appears that the numbers are generated based off a holdout set instead of the training, so I believe the statistics aren’t based off the training data. Also, it appears RapidMiner also optimizes the hyperparameters and builds the best prototypes. Knowing this information, I feel better trusting the results that RapidMiner provided.

Simulating the results of each model with the values of the vehicle we are trying to predict the rating for resulted contradicting conclusions. I first simulated the vehicle on the Gradient Boosted Tree model which provided a 82% probability of an acceptable rating. The safety, persons, and luggage helped this rating, while the very high maintenance costs offered a level of uncertainty. The Support Vector Machine model produced a 63% probability of an unacceptable vehicle rating. Attributes that had a positive impact in support of an acceptable rating for the Gradient Boosted Tree model had a negative affect for the SVM. Deep Learning results were like the Gradient Boosted Tree’s results where maintenance cost drove down the likelihood of receiving an acceptable rating, but still predicted an acceptable rating at a probability of 52%.

To conclude this analysis, I would not recommend offering the customer the vehicle. The top models RapidMiner produced indicated a wavering possibility that the customer would rate the vehicle as acceptable at a threshold of 75%. Instead, we should think about offering vehicles that are more likely to produce an acceptable rating such as those with a larger luggage boot, lower maintenance costs, can fit more people, lower purchase price, and higher safety. Those features have a higher correlation with an unacceptable rating.